Spectrum Profi Club

für alle Spectrum und SAM Freunde



Wolfgang und Monika Haller
Ernastr. 33, 51069 Köln, Tel. 0221/685946
Bankverbindung: Dellbrücker Volksbank
BLZ 370 604 26, Konto-Nr. 7404 172 012

Ausgabe 75

März 1996

Geschafft!

Unsere 75. Ausgabe steht! Und nachdem Moni fürchterlich (!) geschimpft hatte (wegen des späten Erscheinungstermins der letzten Ausgabe), habe ich mal so richtig Gas gegeben. Mit Erfolg! Somit erscheinen wir diesmal wieder zeitig, trotz kurzem Monats, und - da es ja unsere Jubiläumsausgabe ist - auch noch mit 4 Seiten mehr als üblich.
Dabei kommen die SAM-User diesmal schlecht weg. Es ist einfach so gut wie nichts bei uns eingegangen. Auch wir sind noch in der Festplatten-Testphase, sodaß unser Artikel dazu wohl erst im kommenden Monat erscheint. Was ist los mit euch? Steht der SAM nur zur Dekoration rum? Also, ran an die Tasten... alles ist von Interesse. Denkt dran, es ist euer Info von Usern für User!
Zur DTP-Hilfe kam bisher auch noch keine Frage. Dafür aber aufmunternde Briefe, das ich mal einen Einstiegs-Artikel schreiben soll. Hm - das ließe sich machen, aber Fragen von euch würden mir den Einstieg in einen solchen Artikel enorm erleichtern.
Da viele von euch offensichtlich gerne mal etwas fürs Info schreiben möchten, ihnen aber kein Thema einfällt, hier ein paar Entscheidungshilfen. Es handelt sich um die Fortsetzung zum Thema unserer Postkartenaktion: "Was vermißt ihr im Info":
Software und Freesoft-Tests (!), Spielelösungen/-tips nicht nur vom Adventures, z.B. von Ultimate-Programmen, MC- und C-Programmierkurs, ROM-Routinen (Augen auf, hier im Info), kleine MC-Routinen (nicht näher benannt, welche), Plus D Tips (dito), Beiträge für Anfänger, mehr übers Komprimieren (JPEG, MPEG - eher typische MAC-Formate), Soundformen für Sound- bzw. E-Tracker, Sammeln von Computern, Steuerung von Heizung, Eisenbahn und Robotern, Farbmonitor-Verwendung (welche gehen am Speccy?) und 'Wie sieht es mit dem Anschluß von Magnetplatten (??!) am 48er aus?'.
Und dann ist da noch das immer beliebte Thema Spectrum-Emulatoren (ha, demnächst ein Bericht über einen 128K Spectrum Emulator auf dem MAC!) sowie zur Übertragung von Kassetten-Programmen auf andere Plattformen (PC).
Alfred Pirozzi aus Italien (siehe Mitgliederliste) sucht einen Printerpuffer für seinen GLP II-Drucker am Spectrum 48K.
Zum Schluß noch eine **Antwort von Jean Austermühle** auf die **Frage von Andreas Schönborn** im letzten Info (sie kam 'last minute'):
Ein IF1 kann natürlich irgendwann einmal kaputtgehen, also auch beim Anstecken an den Speccy, aber folgendes sollte bedacht werden: Der Z80 stellt an PIN 27, bzw. PIN 24 oben (Expansions-Bus), das M1-Signal bereit, welches nur für +D, Disciple und IF1 verwendet wird. Bei manchen Z80-Prozessoren fehlt dieses Signal, was aber nur bei der Benutzung von obigen Teilen bemerkt ist, da es vom Spectrum nicht benutzt wird; eine Funktion von Beta-Disc ist somit problemlos möglich.
Das wars für diesesmal. Viel Spaß an diesem Info wünscht euch euer WoMo-Team.

Spectrum FORUM

Ich habe begonnen, eine ZX Spectrum World Wide Web-Seite, genannt **Sinclair ZX Spectrum FORUM** aufzubauen. Diese ist komplett in englischer Sprache abgefaßt. Meine Seite ist unter der Adresse:

http://www-users.informatik.rwth-aachen.de/(tilde)afw/sincfo0.html

zu erreichen. Ich bitte euch und alle anderen Mitglieder um konstruktive Kritik via e-mail, u.a. was die Richtigkeit der technischen Informationen über den ZX Spectrum betrifft. Meine e-mail Adresse lautet:

afw@pool.informatik.rwth-aachen.de

Ich interessiere mich sehr für Bildmaterial, welches ich einscannen und im WWW anbieten könnte. Ich suche vor allem Aufnahmen vom ZX Spectrum und Spectrum+, sowie Interface 1, Microdrive, Diskettenlaufwerke etc. Auch bin ich sehr an allen Informationen über Aufstieg und Fall der Sinclair Ltd., sowie technischen Details der Sinclair-Hardware interessiert. Das zugesandte Material bekommt ihr umgehend wieder zurück.

Alexander Walz, App. 1203
Kastanienweg 4-6, 52074 Aachen

Basicode 2 lebt!!!

Sensation aus den Niederlanden.
Nach einer fast 2-jährigen Odyssee auf der Suche nach einer SAVE-Routine für den ZX 81 im "Basicode 2 Modus" bin ich nun dank zweier sympathischer ZX81-User aus Holland fündig geworden.
Nach einer Information von Herrn Leo Moll hat Herr Jack Raets eine "Basicode 2 Save-Routine" für den ZX81 geschrieben. Telefonisch hat mir dies Herr Raets bestätigt, er hat die SAVE-Routne bereits am ZX81 und am PC erfolgreich getestet. Damit ergibt sich erstmals die Chance, daß der ZX81 aufschließt und über "Basicode 2" voll kompatibel zu den Computern: Aplle II + IIe, BBC Model A+B, Colour Genie, Commodore 3000, 4000 und 800, C64, PET 2001, VC20, DAI, Dragon 32, Junior (9K Version), Junior Standard, Junior mit VDU-Karte, Sharp MZ80A, Sharp MZ30K, **ZX SPECTRUM**, TRS 80 Modell I + III sowie Videogenie und vielleicht zum KC 85 wird.
Ich habe mich über diese Chance sehr gefreut und bedanke mich herzlich bei den beiden ZX-Usern aus den Niederlanden.

Henning Räder, Emmericher Straße 35
46147 Oberhausen, Tel. 0208/688969

Die neue Mitglieder-liste 1996



Erwin Müller
Strehlener Str. 6b, 01069 Dresden
Heinz Schober
Taubenheimer Str. 18, 01324 Dresden
Detlef Witek
Am Bornberg 16, 04435 Schkeuditz
Jens Mückenheim
Heidelbergstr. 20, 06577 Braunsroda
Uwe Finke
Fritz-Reuter-Str. 6, 06766 Wolfen
Gerd Schibelius
Mühlbecker Weg 4, 06774 Pouch
Ingolf Fitzner
Talstraße 7, 07407 Rudolstadt
Stefan Zimmermann
Friedenssiedlung 6, 07629 Hermsdorf
Ilja Friedel
Schrödingerstr. 10, 07745 Jena
Klaus Peschke
Rosa-Luxemburg-Str. 43, 08112 Wilkau-Haßlau
Albert Wolter
Kieselhausenstr. 23g, 09117 Chemnitz

Hans Schmidt
Fredersdorfer Str. 10, 10243 Berlin
Eduard Bröse
Lindenstr. 34-35, Zi 56, 10969 Berlin
Hans-Christof Tuchen
Lotzestr. 10, 12205 Berlin
Frank-Michael Moczko
Haselsteig 41, 12347 Berlin
Ingo Wesenack
Spandauer Damm 140/10, 14050 Berlin
Hans-Jürgen Klawiter
Belziger Str. 20, 14823 Niemegk
Karl-Heinz Germeck
Rote Kapelle 1, 15230 Frankfurt/Oder
Wilko Schröter
Pestalozzistraße 38, 17438 Wolgast
Wolfgang Krille
Gaußstr. 6a, 17491 Greifswald

Carsten Pfeil
Mittlerer Landweg 226, 21035 Hamburg
Michael Horrer
Sommerhuderstr. 13, 22769 Hamburg
Willi Mannertz
Lindenstr. 12, 24223 Raisdorf
Holger Dittmann
Burgstr. 28, 24939 Flensburg
Günther Marten
Staulinie 12, 26122 Oldenburg

Wolf-Dietrich Lübeck
Jakobistr. 15, 30163 Hannover
Mustafa Knobel
Schubertstraße 4, 30823 Garbsen
Frank Schlüter
Pehlen 2b, 32108 Bad Salzuflen
Guido Schell
Auf dem Stocke 37, 32584 Löhne
Stephan Preuß
Banater Str. 4, 32832 Augustdorf
Dieter Hucke
Korbacherstr. 241, 34132 Kassel
Dirk Berghöfer
Am Kalkrain 1, 34549 Edertal-Giflitz
Nele Abels
Ketzerbach 57, 35037 Marburg
Matthias Wiedey
Grünberger Str. 190, 35394 Giessen
Horst Engelhardt
Im Eisenbach 5, 35716 Dietzhölztal
Bernd Kalla
Robert-Koch-Str. 3, 36043 Fulda
Slawomir Grodkowski
Wolfgang-Döring-Str. 11, 37077 Göttingen
Günter Hartwig
Lissabonstr. 14, 37079 Göttingen
Hartmut Käsemann
Ahnewende 14, 37191 Katlenburg-Lindau
Peter Liebert-Adelt
Lützowstr. 3, 38102 Braunschweig
Emil Obermayr
Teichmüllerstr. 2, 38114 Braunschweig
Christian Scharmberg
Olvenstedter Grund 27, 39130 Magdeburg

Michael Meyer
Halskestr. 6, 40215 Düsseldorf
Roland Kober
Josef-Neuberger-Str. 42, 40625 Düsseldorf
Martin Pollok
Ina-Seidel-Str. 29, 40885 Ratingen
Peter Rennefeld
Genhodder 19, 41179 Mönchengladbach
Jean Austermühle
Postfach 10-1432, 41546 Kaarst
Holger Langheim
Hastener Str. 69, 42855 Remscheid
Andreas Schönborn
Gössingstr. 44, 44319 Dortmund
Peter Miosga
Holtbredde 11, 45711 Datteln
Frank Dopierala
Im Eickelkamp 158, 47169 Duisburg
Bodo Schulte-Varendorff
Kiefernweg 3a, 49090 Osnabrück

Frank Meurer
Schulstr. 21, 50389 Wesseling
Roland Kaiser
Am Trutzenberg 44, 50676 Köln
Holger Hanewacker
Gellertstr. 25, 50733 Köln
Lothar Ebelshäuser
Grasegger Str. 49, 50737 Köln
Monika und Wolfgang Haller
Ernastr. 33, 51069 Köln
Stephan Haller
Broicher Str. 60, 51429 Bergisch Gladbach
Rudolf Herzog
Goldbornstr. 63, 51469 Bergisch Gladbach
Alexander Walz
Kastanienweg 4-6, 52074 Aachen
Stefan Kriebs
Blachweg 31, 52355 Düren
Wilhelm Dikomey
Mühlengasse 24, 52391 Vettweiß
Siegfried Dikomey
Brunnenstr. 41, 52531 Übach-Palenberg
Ian D. Spencer
Fichtenweg 10c, 53804 Much
Paul Webranitz
Borgasse 16, 54538 Kinheim
Josef Menzel
Dr. Gebauer Str. 71a, 55411 Bingen
Michael Heckelei
Im Stübbeken 33, 58642 Iserlohn

Hans-Joachim Blume
Kohlrauschweg 19, 60486 Frankfurt
Wolfgang Berndt
Friedberger Str. 92c, 61169 Friedberg
Dragan Pavlovic
Buchenweg 8, 63667 Nidda
Friedrich D. Mehedinti
Burgstr. 51, 67659 Kaiserslautern
Markus Stieglitz
Schmitzstr. 4, 68219 Mannheim
Günter Berg
Hölderlinstr. 38, 68259 Mannheim
Peter Bergmann
Reiherstr. 30, 68309 Mannheim

Frederic Dürkes
Mezgerstr. 43, 70563 Stuttgart
Günter Keefer
Erzgebirgeweg 16/1, 70736 Fellbach
Thomas Eberle
Gastäckerstr. 23, 70794 Filderstadt
Werner Wagner
Alb.-Einstein-Str. 3, 71364 Winnenden
Christof Odenthal
Schopenhauerstr. 36, 73431 Aalen
Helge Keller
Herm.-Löns-Weg 51, 76307 Karlsbad

Dr. Martin Weltzer
Ebenbödestr. 4, 81241 München
Stefan Ballerstaller
Berg am Laim Str. 146, 81673 München
Wolfgang Gierisch
Habichtstr. 9, 82223 Eichenau
Claus-Jörg Weiske
Veit-Stoß-Str. 2, 82256 Fürstenfeldbruck
Eckart Reich, c/o R. Helbing
Dom.-Ringeisenweg 3, 82380 Peißenberg
Harald R. Lack
Heidenauerstr. 5, 83064 Raubling
Ludwig Halbritter
Voglerinweg 7, 86508 Rehling
Herbert Hartig
Postfach 323, 86803 Buchloe
Wolfgang Rapp
Säntisstraße 4, 88145 Hergatz
Roland Albert
Jahnstraße 3, 88662 Überlingen/Bodensee
Thomas Schwarz
Raiffeisenstr. 9, 88969 Owingen
Dieter Münz
Iglauer Str. 44, 89518 Heidenheim

Michael Jarowy
Dr. Carlo Schmid Str. 150, 90491 Nürnberg
Günter Brütting
Waidacher Dorfstr. 34, 91278 Pottenstein
Rupert Hoffmann
Sonnenstr. 2, 92637 Weiden
Nico Kaiser
Geschw.-Scholl-Str. 11a, 98693 Ilmenau
Scott-Falk Hühn
Offenhainer Str. 4, 99610 Sömmerda

Unsere ausländischen Mitglieder

ÖSTERREICH:
Leszek Chmielewski Daniel
Prager Straße 92/11/12, A-1210 Wien
Walter Sperl
Uferstr. 308, A-2625 Schwarzau/Stfd.
Peter Meindl
Siemensgasse 3/8, A-2630 Ternitz
Georg Gojcevic
Badweg 6, A-6923 Lauterach

ARGENTINIEN:
Dierk Reuter - Colegio Pestalozzi
Freire 1824, ARG-1428 Buenos Aires

SCHWEIZ:
Toni Arpagaus
Zwärenstr. 8, CH-4118 Rodersdorf
Willi Stalder
Kriegstettenstr. 24, CH-4553 Subingen

DÄNEMARK:
Preben D. Sörensen
Jaegersgardsgade 128, DK-8000 Arhus C.
Arne Nielsen
Chr X's vej 10st, DK-8260 Viby J

GROSSBRITANNIEN:
Andy Davis - 62 Tithe Barn Lane
Woodhouse, Sheffield, S13 7LN, England
Dave Fountain - 11 Camel Road
Silvertown, London, E16 2DE, England
Miles Kinloch - Flat 16, 6 Drummond Street
Edinburgh, EH8 9TU, Scotland, U.K.

ITALIEN:
Alfred Pirozzi
Pfarranger Weg 5, I-39049 Sterzing

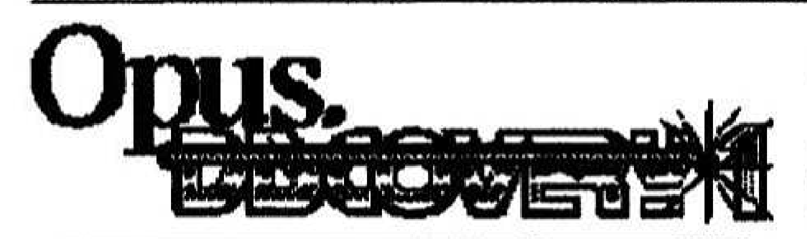
NIEDERLANDE:
Rob Roggeveen
Waterlustlaan 87, NL-2804 KX Gouda
Roelof Koning
Selwerderstr. 26, NL-9717 GK Groningen
Johan Koning
Wieden 6, NL-9866 TM Lutjegast

POLEN:
Ed Polinski
Boczna 27, PL-05-300 Minsk Maz

RUMÄNIEN:
Marin Stanculescu - Str. Ciuruleasa 3
Sect. 4, O.P.7, ROM-75445 Bucuresti

Diese Liste ist mit Ausnahme Großbritanniens nach Postleitzahlen geordnet, damit man sehen kann, ob man einen "Gleichgesinnten in direkter Nachbarschaft hat.

Stand: 24. Februar 1996
Mitgliederzahl: 115



Wie versprochen geht es heute darum, wie wir die Header der einzelnen Sektoren lesen können. Im Prinzip funktioniert es genauso, wie das Lesen des Datenbereichs. Wir können also die alte Routine "lesen" aus dem letzten Heft verwenden. Der Unterschied besteht nur im Aufruf, also dem Wert, mit welchem das Register A geladen wird. Die Routine kann dann wie folgt aussehen:

Sektorheader laden

```
LD HL,lesen        ;Adresse der Laderoutine
LD BC,6            ;der Header enthält 6 Bytes
LD DE,Buffer       ;ab der Adresse "Buffer"
                   ;werden die Headerdaten
                   ;im Speicher abgelegt
LD A,192+8*"Keine Anlaufphase"
     +4*"Verzögerung"
CALL WAIT          ;Routine aufrufen
JR NZ,error        ;wenn Z-Flag=1, trat Fehler
                   ;auf
```

Wird im Register A das Bit 3 gesetzt (entspricht der Addition von 8 zu 192), wird auch bei stillstehender Disk sofort versucht, den folgenden Sektorheader zu lesen - nicht besonders sinnvoll! Wird Bit 3 auf 0 gelassen, so wird bei stillstehender Disk zuerst der Motor angeworfen und 6 Umdrehungen abgewartet. Dann wird der Befehl ausgeführt. Wird Bit 2 in A gesetzt (also 4 addiert), wartet der Controller 30ms und führt den Befehl erst dann aus. Dies ist normalerweise nicht nötig, nur bei sehr alten Laufwerken wackelt der Kopf nach Stepbewegungen so stark nach, daß diese Beruhigungsphase nötig ist.

Wurden die Headerdaten korrekt gelesen, finden sich im Speicher ab Adresse "Buffer" folgende Informationen:

Tracknummer (1 Byte):
Tracks sind von 0 ab nummeriert
Seitennummer (1 Byte): 0/1
Sektornummer (1 Byte):
Bei Opusdisketten von 0 an, bei +D- und IBM-Disks von 1 an nummeriert
Sektorlänge (1 Byte):
0=128, 1=256, 2=512, 3=1024 Bytes
CRC-Bytes (2 Bytes): Cyclic Redundancy Check

Wozu können wir diese Informationen verwenden?

Die Möglichkeit, Sektorheaderdaten zu lesen, eröffnet uns viele Möglichkeiten, unbekannte Disketten zu erforschen oder auch schnelle Routinen für Opusdisks zu schreiben. So sind z.B. auf normalen Opusdisketten (mit dem Standard-ROM) die Sektoren auf einem Track nicht einfach in der Reihenfolge 0,1,2,...,17 abgelegt. Da die Lese- und Schreibroutinen im Opus-ROM sehr langsam sind (unter anderem, weil sie viele Teile der Cassettenladeroutinen im Spectrum-ROM verwenden), liegen zwischen zwei logisch aufeinanderfolgenden Sektoren (z.B. 3 und 4) zwölf andere Sektoren. Man spricht daher von einem Interleave der Größe 13. Wieso 13? Stellen wir uns die Sektoren als nebeneinanderliegende Fächer vor. Das erste Fach habe die Nummer 3 und wir suchen jetzt Fach 4. Dazu müssen wird 13 mal nach rechts gehen, daher Interleave 13.

Wenn wir Lese- und Schreibroutinen selbst programmieren, können wir sogar direkt aufeinanderfolgende Sektoren lesen und schreiben, es ist also ein Interleave von 1 möglich. Dies ist auch im Quick-ROM für die Befehle LOAD, SAVE und MERGE realisiert. Meine zwei Programme "Quick-Copy" und "Quick-Move" verwenden auch diese Technik, um bis zu sieben mal schneller alsdie normalen Opus-Befehle arbeiten zu können. Doch wie erreichen wir dies?

Dazu müssen wir die Anordnung der einzelnen Sektoren auf einem Track kennen. Normalerweise sind alle Tracks gleich aufgebaut, so daß eine Analyse des ersten Tracks genügt. Wir gehen wie folgt vor: Ansteuern des Tracks 0 und Auswahl der Seite 0 (damit die Routine auch sicher bei einseitigen Disks funktioniert). Dann lesen wir den ersten Sektorheader ein und speichern seine Sektornummer ab. Nun lesen wir weitere Sektorheader ein und speichern deren Nummer in unserer Liste. Wenn die erste gefundene Nummer erneut auftaucht, sind wir fertig. Bei einer Standardopusdisk haben wir jetzt 18 Sektornummern eingelesen, nämlich die Zahlen 0 bis 17. Bei einer mit dem Quick-ROM formatierten Disketten sind sie in ihrer natürlichen Reihenfolge, wobei aber nicht unbedingt mit Nummer 0 begonnen wird. Die Liste könnte also wie folgt aussehen:

5,6,...,16,17,0,1,2,3,4

Bei einer mit dem Standard-ROM formatierten Disk sieht die Liste vielleicht wie folgt aus:

7,14,3,10,17,6,13,2,9,16,5,12,1,8,15,4,11,0

Jetzt könnten wir mit einer kleinen Leseroutine die Sektoren genau in dieser Reihenfolge einlesen und dann auf eine zweite Diskette schreiben. Genau dies macht "Quick-Copy" beim Kopieren von ganzen Disketten, wobei obige Listen sowohl für die Quell- wie die Zieldiskette angelegt werden. Dadurch kann unter anderem auch eine Stan-

dard-Disk auf eine Quick-Disk kopiert werden. Die Lese- und Schreibroutinen müssen nur genau darauf achten, wohin sie im Speicher die Sektordaten ablegen, damit hier kein Chaos entstehen kann. Doch jetzt noch die Routine, um die Liste zu erzeugen:

```
start   CALL 5896     ;Opus einschalten
        DI            ;Interrupts aus
        LD HL,12288   ;Laufwerks- und
                      ;Seitenregister
        LD A,(HL)     ;alte Daten lesen
        AND 236       ;Seite 0 wählen und
                      ;Laufwerksnummer
                      ;löschen
        OR 1          ;Laufwerk 1 auswählen
        LD (HL),A     ;Informationen speichern
        XOR A         ;A=0, kann auch 1-3 sein
                      ;(=Steprate)
        CALL WAIT     ;Kopf auf Track 0
                      ;fahren
        LD IX,Sek_Nr  ;Adresse der Liste im
                      ;Speicher
        CALL SEKNR    ;Sektornummern lesen
        EI            ;Interrupts ein
        JP 5960       ;Opus aus und zurück ins
                      ;BASIC
SEKNR   PUSH IX       ;Startadresse Liste
                      ;sichern
        CALL SEKNR1   ;erste Sektornummer
                      ;lesen
SEKNR2  PUSH AF       ;erste Sektornummer
                      ;sichern
        CALL SEKNR1   ;nächste Nummer lesen
        LD B,A
        POP AF        ;erste Nummer holen
        CP B          ;Sektornummer erneut
                      ;gelesen?
        JR NZ,SEKNR2  ;nein, dann weiterlesen
        PUSH IX       ;aktuelle Listenadresse
                      ;auf Stapel
        POP HL        ;und dann in HL holen
        POP BC        ;erste Listenadresse
        SCF
        SBC HL,BC     ;Anzahl der Sektoren
                      ;bestimmen und
        PUSH HL       ;im Registerpaar BC ins
                      ;BASIC
        POP BC        ;übergeben
        RET
SEKBUF  DEFS 3        ;3 Bytes Platz für
                      ;Sektorheaderdaten
SEKNR1  LD DE,SEKBUF  ;hierhin Daten laden
        LD HL,lesen   ;unsere bekannte
                      ;Leseroutine
        LD BC,3       ;nur die ersten 3 Bytes
                      ;lesen
        LD A,192      ;Befehl: Header lesen
        CALL WAIT     ;unsere Routine
        DEC DE        ;Zeiger auf Sektor-
                      ;nummer
        LD A,(DE)     ;Sektornummer in Liste
                      ;speichern
        LD (IX+0),A
        INC IX        ;nächste Listenadresse
        RET
```

In der obigen Form speichert die Routine die erste Sektornummer zwar nochmals am Ende der Liste, braucht also bei einer Standarddisk 19 Bytes Listenplatz, ist dafür aber sehr einfach zu verstehen. Mit dem Aufruf

```
LET sekanzahl=USR start
```

erhalten wir die Anzahl der Sektoren auf Track 0 der eingelegten Disk zurück und die Nummern aller Sektoren in unserer Liste. Falls ein Fehler beim Einlesen auftrat, merkt dies die Routine in ihrer jetzigen Form nicht, aber es sollte kein Problem für Euch sein, dies einzubauen.

Soviel für heute, viel Spaß und Nutzen mit den Routinen. Anmerkungen, Fragen, Kritik und Lob wie immer an:

Helge Keller, Hermann-Löns-Weg 51
76307 Karlsbad, Tel.:07202/6076

EINZEILER

Auch heute wieder zwei Einzeiler aus der ehemaligen "Sinclair Gids". Der erste stammt von D. de Gier, der zweite Autor ist uns nicht bekannt. Viel Spaß beim Eintippen und Ausprobieren.

```
10 LET q=3600: LET r=65536: LET t$=
":0": DEF FN m()=INT ((r*PEEK 23674+
256*PEEK 23673+PEEK 23672)/50): LET m=
FN m(): LET n=FN m(): LET o=m<24*q:
LET n=o*(m+n+ABS (m-n))/2: LET u=INT
(n/q): LET m=INT ((n-u*q)/60): LET s=
n-u*q-m*60: PRINT AT 0,24;t$(2+(u>9)
TO );u;t$( TO 2-(m>9));m;t$( TO 2-(s>9
));s: LET p=1-o+n: LET n=(u+(INKEY$=
"u")-(INKEY$="U"))*q+(m+(INKEY$="m")-
(INKEY$="M"))*60+(s+(INKEY$="s")-
(INKEY$="S")): LET o=n<>p: LET n=ABS n
*50: LET p=INT (n/r): LET q=INT ((n-p*
r)/256): LET r=n-p*r-q*256: POKE 23674
*o,p: POKE 23673*o,q: POKE 23672*o,r:
GO TO 10
```

```
10 FOR a=30000 TO 30018: READ b: POKE
a,b: NEXT a: PRINT "SPECTRUM PROFI
CLUB": PAUSE 66: RANDOMIZE USR 30000:
RESTORE : GO TO 10: DATA 33,0,64,78,
6,8,203,17,31,16,251,119,35,124,254,88,
32,241,201
```

Cracking Workshop

TEIL 3

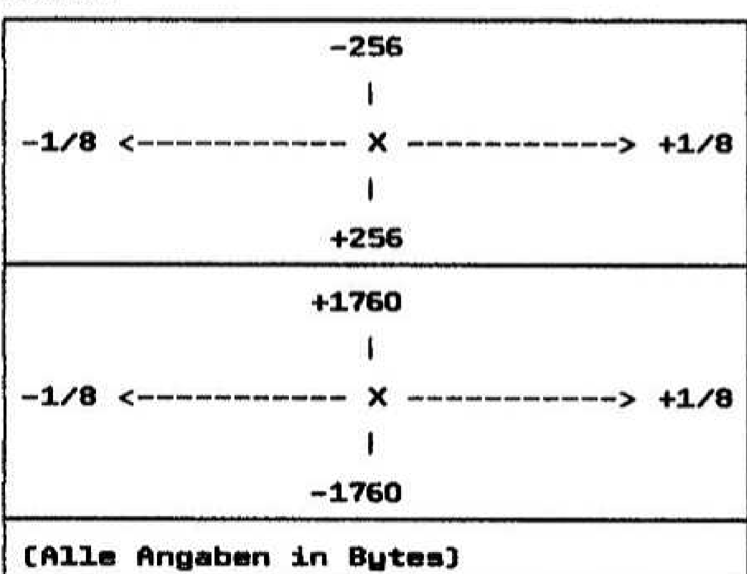
In diesem Monat widme ich mich der Verschönerung von Progs, die mit LERM Conversion Utility gecrackt wurden und den Programmen, welche als ganzer Speicherinhalt geladen werden.
LERM CU6 ist eigentlich ein ganz nützliches Programm, denn er crackt automatisch einige sehr verzwickte Systeme wie Speedlock I, II, III, Firebird und Alcatraz, leider sieht der Screen beim Laden von Alcatraz Programmen ziemlich kaputt aus. Die Lösung, wie man es menschenwürdiger machen kann ist ganz einfach.
Ich nehme immer beim Laden des Orginalprogramms einen Snap vom Screen und verwende ihn beim Laden. Der LERM Crack wird nach dem Laden des Programms zwar wiederhergestellt, aber während des Ladens sieht es ziemlich blöd aus, deshalb kompressiere ich immer den Hauptteil und hänge den zerstörten Screen nach dem Codeteil dazu, Platz müßte ja genug sein, ich könnte aber auch noch den Screencompressor Plus verwenden um den Screen noch zu packen. Auch wenn das Ergebnis nicht immer zufriedenstellend ist, braucht man keine weiteren Werkzeuge.
Nach dem Laden des Hauptteiles dekompressiere ich dann den Screen, der das schöne Titelbild überschreibt. Auf der Disk befindet sich auch ein Programm namens LDIR mit dem man eine Speicherverschiebung ganz leicht in MC designen kann, damit läßt sich auch ein Bild aus dem Speicher hervorholen.
Nachdem der Screen also so ist wie es ursprünglich sein sollte, dekompressiere ich den Hauptcodeteil und starte das Programm mit USR 16384. Das Endergebnis sieht dann so aus, daß nach dem Laden des Programms das Titelbild nur für eine Sekunde zerstört wird und dann wieder zusammengesetzt wird. Das Programm kann dannach mit einem Knopfdruck gestartet werden. Auf diese Weise habe ich MAZEMANIA von YS Covertape gecrackt, und noch viele andere.
Ein weiterer Vorteil ist das nur drei Files übrigbleiben, statt wie üblich vier. Bei +D, bei dem pro Disk nur 80 Files draufpassen ist das schon ein gutes Ergebnis und weil die Files gepackt sind, dauert es etwas kürzer, sie von Tape zu laden und es passen auch mehr Programme auf eine Kassette.
Bedeutend mehr Arbeit habe ich investiert um Programme wie 3D STARSTRIKE oder DEFENDER OF THE CROWN zu cracken. Ohne das hervorragende polnische Kopierprogramm COPY COPY hätte ich es aber wohl nie geschafft.
Das Problem war, das die Files zu lang sind um noch eine Routine im Speicher unterzubringen. Die Programme belegen meistens 49152 Bytes und das ist nunmal der gesamte Speicherplatz des Spectrum 48K und auch ein 128K Spectrum hilft da nicht weiter. COPY COPY bietet die Funktion LOAD (x TO y), näheres dazu aber in der Anleitung, die sich in der Freewarebibliothek von ASC befindet.
Diese Funktion erlaubt lange Programme zu zerschneiden und dadurch erleichtert sie das cracken von solch langen Programmen. Mit Screencompressor+ lade ich zuerst das Titelbild, das Programm stoppt automatisch, wenn 6912 Bytes eingelesen wurden (Bei CAULDRON II ist es daher unnützlich, aber da funktionierte wenigstens das SNAPOUT, siehe SCENE 12/95). Als nächstes wird mit COPY COPY nach 6912 Bytes ein 1704 Bytes langes Teil isoliert, das am Ende wieder ab 23296 stehen muß. Es ist leicht zu errechnen, das der nächste Teil nach 8616 Bytes gesplitet werden muß, diesesmal bis zum Ende, also 40536 Bytes lang. Dieses Teil wird dann im Speicher ab der Adresse 25000 stehen, das läßt Platz übrig für "Cracked By ..." Message.
Das andere Cracking Utility das ich auf die Disk abgespeichert habe, heißt LDHDLS (Load Headerless), damit ist es möglich, die so gewonnenen Teile von Tape zu laden um sie auf Disk zu speichern.
Der Hauptteil (CODE 25000, 40536) muß kompressiert werden, entweder mit TURBO IMPLODER, was etwas unsicher ist, aber keine weiteren Nachteile mit sich bringt, oder PK LITE, dessen Dekompressationsadresse aber im Screenbereich liegen muß, was das Titelbild etwas verunstaltet.
Die ermittelte Startadresse gibt man als Execution ein, daher besteht die Möglichkeit, das Codeteil zu Verifizieren, nur dann wenn man zur Probe ohne die Executeadresse kompressiert, dekompressiert und mit Ursprungscode verifiziert (+D User, es besteht die Möglichkeit, VERIFY pn zu benutzen).
Nehmen wir jetzt an, das Codeteil wurde von 40536 auf 32411 Bytes kompressiert, in diesem Fall ist das zweite Codeteil mit der Länge von 1704 Bytes auf die Adresse 57411 zu laden, dann haben wir ab der Adresse 59115 freien Speicherplatz, aber nicht so schnell, denn irgendwie muß dieses Codeteil wieder auf die richtige Adresse gebracht werden, also 23296. Dafür können wir wieder das Programm LDIR verwenden (Source=57411, Length=1704, Destination=23296, Start=25000, Art des Sprungs=195, Startadr.=59115). Das so erzeugte MC muß mit dem Hauptcodeteil abgespeichert werden, der von Loader erfahrene SP Wert ist von Fall zu Fall verschieden, muß aber auch eingegeben werden, sonst stürzt das Programm ab.

Nach RAND USR 59115 (wieder theoretisch, in diesem Fall), wird zuerst der STACK POINTER auf den richtigen Wert gesetzt, dann kopiert das Programm die 1704 Bytes von 57411 nach 23296 und ruft die Dekompressationsroutine auf, die wiederum dann das Programm automatisch aufruft.
Es ist zwar eine Quälerei, aber das Ergebnis entschädigt für die Mühe und der Crack ist viel sauberer als manch ein anderer. Auf diese Weise habe ich STARSTRIKE II, WANDERER 3D und viele andere Programme gecrackt, während anderswo immer noch MF "Cracks" kursieren.
So, das wärs eigentlich vorläufig. Die Serie wird fortgeführt, wenn weitere Fragen zu dem Thema eintreffen, oder wenn ich wieder mal was neues zu berichten finde, denn bis jetzt habe ich fast meinen gesamten Erfahrungsschatz bezüglich Cracks veröffentlicht.

L.C.D./Tiger's Claw

Der Bildspeicher des ZX Spectrum: keine Angst!

Wer einmal auf die Idee kommt, in Maschinencode das zu programmieren, was die Basic-Befehle PLOT und DRAW tun, ohne dazu die Spectrum ROM-Routinen zu nutzen, der merkt sofort, daß der Bildspeicher, das Video-RAM, kompliziert aufgebaut ist. Aber dennoch ist natürlich alles logisch und soll hier einmal erläutert werden.
Der Spectrum Screen besteht aus 24 Zeilen zu je 32 Spalten, also aus 768 Zeichenpositionen. Jede davon besteht aus 8 x 8, also 64 Pixeln, das macht insgesamt 49152 Bits oder 6144 Bytes. Anders gerechnet: Der Spectrum kann 256 x 192 Pixel auflösen, macht auch 49152 Bits. Diese Bytes sind im Spectrum-RAM ab Adresse 16384 abgelegt und gehen bis zur Adresse 22527. Das ist immer fest. Da für jedes Pixel genau ein Bit eingeplant ist, kann man also hier keine Farbinformation ablegen, dies geschieht im Attributspeicher und wird weiter unten erläutert.
Welches Pixel steht aber nun an welcher RAM-Adresse? Leider sind die Bildschirmbits (0 für nicht gesetzt, 1 für gesetzt) nicht einfach Zeile für Zeile hintereinander im RAM abgelegt, sondern in anderer Reihenfolge! Der Screen besteht zunächst einmal aus 3 Teilen (oberer, mittlerer und unterer Teil). Jedes Drittel hat natürlich 8 Zeilen mit je 8 Pixeln, weil wir insgesamt 24 Zeilen haben. Als erstes steht ab RAM-Adresse 16384 die oberste Pixelzeile des 1. Drittels, also 32 x 8 = 256 Bits oder 32 hintereinanderfolgende Bytes. Danach kommt die erste Pixelzeile der nächsten Zeichenreihe des oberen Drittels des Screens, also versetzt um 32 Bytes. Dann ist die erste Pixelreihe der nächsten Zeichenreihe dran und so fort. Zwei untereinanderliegende Pixelzeilen sind also 32 Bytes (je Zeichenreihe) mal 8 Zeichenreihen = 256 Bytes im RAM auseinander.
Nebeneinanderliegende Pixel sind genau ein Bit auseinander. Nur wenn die Zeichenreihe bei übereinanderstehenden Pixeln wechselt, errechnet sich der Versatz des in die nächste Zeichenreihe fallenden Pixels zu 8 x 32 x 7 (RAM der untersten Pixelzeile) minus 32 (RAM der obersten Pixelzeile sind in zwei Zeichenreihen um 32 Bytes auseinander). Das ergibt ein einfaches Schema:



(Alle Angaben in Bytes)

Für die Errechnung der RAM-Adresse kann man eine einfache Formel angeben, wenn man die Position auf dem Screen beschreibt mit Zeilennummer (ZNR), Pixelzeile in der Zeichenzeile (PNR) und Pixelspalte (0-255) PSP:
RAM = ZNR*32 + PNR*256 + PSP/8 + 16384
Für das zweite bzw. das letzte Bildschirmdrittel muß 16384 ersetzt werden durch: 19432 bzw. 20480 (jeweils 6144/3 = 2048 addiert).
Ein Beispiel: die unterste Pixellinie im letzten Drittel hat die Koordinaten ZNR = 7, PNR = 7. Das letzte (rechte) Pixel ist PSP = 255. In die Formel eingesetzt ergibt: 22527. Das ist genau das letzte Byte des Bildschirm-RAM!
Das kann man in Maschinencode-Programmen gut verwenden. Beispiele sind Text- und Spritescroller. Da muß ein Pixel ja um ein oder mehrere Positionen nach oben, unten, rechts oder links verschoben werden. Speichert man die Position des Sprites in den oben genannten Koordinaten, kann der Spectrum sehr flott die RAM-Adresse des benachbarten Pixel berechnen. Man spart Zeit gegenüber der einfacheren Darstellung in den Angaben PLOT-Position (0-255, 0-191).
Der Attributspeicher wird im zweiten Teil des Artikels erklärt.

Andreas Schönborn, Gössingstraße 44
44319 Dortmund, 9.2.1996

EINE REISE INS SPECTRUM-ROM

Aus dem englischen übersetzt von Stefan Ballerstaller, veröffentlicht mit freundlicher Genehmigung von Andy Davis/Alchemist Res.

Das Spectrum-ROM enthält eine ganze Menge brauchbarer Routinen und Funktionen. Einige von ihnen sind für BASIC-Programme nützlich - z.B. die Routine für Screen abrollen - während andere Zeit und Mühe ersparen können, während man in Maschinencode programmiert.
In diesem Artikel werde ich darlegen, was einige der nützlichen Routinen leisten können und Einzelheiten nennen, wie man sie anwenden kann. Es gibt Situationen, wenn MC-Programmierer BASIC-Funktionen vervielfältigen muß und deshalb habe ich auch einige davon mit berücksichtigt. Nun, fangen wir von vorne an, wie man so schön sagt!

00000 (0000h) USR 0 RST 0

Dorthin gelangt der Spectrum, wenn Sie ihn einschalten - er leert den ganzen Speicher, baut die Bildschirmfarben auf und richtet alle Systemvariablen ein. Schließlich zeigt er das Titelbild. Nicht gerade sehr nützlich!

00008 (0008h) USR 8 RST 8

Der Neustart wegen eines Fehlers. Diese Routine nimmt den Wert des Bytes, das unmittelbar dem RST 8-Befehl folgt, erzeugt die entsprechende Fehlermeldung und übergibt die Kontrolle wieder dem BASIC. Folgt z.B. nach dem RST 8 eine 6, erscheint Fehlermeldung 7, 'Return without GOSUB', und ein Wert von 23 wird die Fehlermeldung '0 Invalid Stream' erzeugen. Dieses Byte ist auch in der Systemvariablen ERRNR bei 23610 gespeichert.

00016 (0010h) USR 16 RST 16

Die PRINT-Routine ist eine der nützlichsten des Spectrum. Ihre Vielseitigkeit läßt sich nur erkennen, wenn sie in MC verwendet wird. Die Routine wird das Zeichen, dessen Zahl im A-Register steht nehmen, und es über die gerade gewählte Leitung (Stream) drucken.
Sie kann Daten über Leitung 3 an einen Drucker oder über Leitung 2 an den Bildschirm schicken, wobei in letzterem Fall die Ausgabe an der gegenwätigen Print-Position liegt und die momentanen Werte für INK, PAPER, FLASH, BRIGHT und OVER verwendet. Diese Routine zerstört allerdings alle Register, vergessen Sie also nicht, sie zuerst auf den Stapel zu setzen!

00040 (0028h) USR 40 RST 40

Dies ist der Neustart für den "Rechner (Calculator)" - er ist äußerst komplex und ich bräuchte diesen ganzen Artikel, um die Grundlagen zu erläutern.

00056 (0038h) USR 56 RST 56

Diese Routine wird 50mal pro Sekunde im Spectrum aufgerufen - sie sucht die Tastatur ab, wobei sie die Systemvariablen ändert und die Systemvariablen der Echtzeituhr in den Adressen 23672, 23673 und 23674 erhöht. Sie holt sich eine abgelegte Adresse vom Stapel (POP), an die sie dann springt.

00654 (028Eh) USR 654

Dies ist die beste Routine zum Absuchen der Tastatur. Sie gibt zwei Werte an die Register D und E zurück. Das D-Register erhält dabei den Wert, der anzeigt, welche Shift-Taste gerade gedrückt wird. Der Wert im E-Register liegt zwischen 0 und 39, gemäß den 40 Tasten der Tastatur.
Wenn beide Shift-Tasten gedrückt werden, hält das D-Register den Code für CAPS-Shift und das E-Register den für Symbol-Shift. Falls keine Taste gedrückt wird, erhält das DE-Register den Wert von FFFFh (65535) zurück.
Wenn mehr als zwei Tasten gedrückt werden oder keine der betätigten Tasten bzw. Tastenpaare eine Shift-Taste ist, wird wieder auf die Nullen zurückgestellt. Die von dieser Routine kommenden Ergebnisse sind recht komplex, doch die folgende Routine ist so gestaltet, daß das Absuchen leichter wird.

00703 (02BFh) USR 703

Dies ruft die Tastaturabsuche-Routine bei 654 auf und gibt den Wert der Taste preis. Der CODE der Taste bzw. Tastenkombination, die gedrückt wurde, wird in der Systemvariablen LASTK (23560) gespeichert. Bit 5 von FLAGS (23611) wird eingerichtet, um anzuzeigen, daß eine neue Taste gedrückt wurde.
Tatsächlich überprüft die Routine den Wert, der in der Systemvariablen MODE steckt, die mitteilt, ob der Computer im K, L, C, E oder G-Modus ist. Dies bewirkt, daß der letzte Wert an LASTK zurückgegeben wird.

00747 (03B5h) USR 747
Dies ist die für den wunderbaren BEEP-Befehl zuständige Routine. In MC sollte das DE-Registerpaar mit einem Wert gleich der Zeit, die die Note (in Sekunden) brauchen wird mal der Tonhöhe geladen werden. Das HL-Registerpaar sollte INT ((437500/Tonhöhe)-30.125) enthalten.
Das klingt sehr kompliziert, und genau das ist es auch. Um für eine Sekunde die mittlere C-Note zu erzeugen, sollte DE INT (261.63•1) und HL INT ((437500/261.63)-30.125) enthalten. Die 30.125 ist eine Konstante, um das Timing des Prozessors zu ersetzen.

01218 (04C2h) USR 1218
ist die SAVE-Routine. Das DE-Register sollte hierbei den Wert der Länge des zu savenden Blocks und das IX-Register die Startadresse enthalten. Register A enthält entweder 0 für einen "Header"-Block oder 255 für einen "Programm/Data"-Block. Das Carry-Flag sollte ebenfalls gesetzt sein.
Diese Routine wird zweimal vom BASIC Save-Befehl aufgerufen, einmal für den 17 Byte langen Header und dann für den Hauptblock der Daten.
Übrigens können Sie den ROM-Code ins RAM kopieren und das Timing ändern, um andere SAVE-Geschwindigkeiten zu erreichen. Die Borderfarben während des Savens können ebenfalls geändert werden.

01366 (0556h) USR 1366
ist die LOAD-Routine. Obwohl für diese Routine genau die selben Register auf exakt die selbe Weise wie für SAVE eingestellt werden müsse, gibt es recht deutliche Unterschiede zwischen ihnen.
LOAD kann bis zu einem bestimmten Ausmaß ziemlich großen Unregelmäßigkeiten der ankommenden Daten folgen. Es spürt hierzu den Rand oder den Punkt an jedem Ausschlag auf, wenn sich der Level des Signals ändert.
Kehrt man von der Routine zurück, wird das Carry-Flag gesetzt, falls der Ladevorgang erfolgreich war, trat ein Fehler auf, wird es auf Null gesetzt.

03282 (0CD2h) USR 3282
die Scroll-Routine. Im Gegensatz zum ZX81 hat der Spectrum keinen eigenen Scroll-Befehl, um das Bild abzurollen. Dies kann von Basic aus mit USR 3282 bewirkt werden. Diese Routine benutzt Herr Spectrum auch, wenn er Sie nach einer Bestätigung nach der Anzeige "scroll?" fragt. Dies ist der wichtigste Anhaltspunkt, später kommen wir auf einige andere, interessante Scroll-Möglichkeiten zurück.

03435 (0D6Bh) USR 3435
die CLS-Routine. Vielleicht nicht sehr nützlich in BASIC, da die Eingabe von CLS leichter ist, doch kann diese Routine in MC hilfreich sein. Die Anzeige wird auf die momentanen Attribute gelöscht (wie sie in den Systemvariablen enthalten sind) und verschiedene diverse interne Funktionen werden über Kanal 'K' ausgeführt.
In Wirklichkeit ist die CLS-Routine des Spectrum unnötig kompliziert und langsam. Eine schnellere und einfachere Art dafür wäre:

```
LD HL,16384  (Start des Display-Files)
LD DE,16385  (Start+1)
LB BC,6144   (Länge der Attribute)
LD (HL),L    (Wert um zu kopieren)
LDIR         (Lösche ihn)
LD BC,792    (Dies löscht
LD A,(8*Paper+Ink)
             die Attribute)
LD (HL),A    (weglassen, um aufzuhören)
LDIR         (Farbe bleibt)
```

Sie können dies verbessern, was aber in den meisten Anwendungen nicht nötig sein wird.

03582 (0DFEh) USR 3582
eine weitere Scroll-Routine. Wird diese Routine bei 3582 gestartet, rollen alle 23 Zeilen des Bildschirmes nacheinander nach oben. Auch Attribute werden beachtet. Diese Routine wird mit USR 3282 selbst aufgerufen.
Wenn Sie die Routine zwei Bytes weiter aufrufen, können Sie unten einen Absatz wählen, der abgerollt werden soll. Falls sie im B-Register mit - sagen wir - dem Wert 10 einsteigen, wird fortwährender Aufruf von USR 3584 nur die unteren 10 Zeilen nach oben rollen, der Rest der Anzeige darüber bleibt unverändert.
Diese Methode wurde in einigen Adventures angewandt, um die INPUT-Zeilen abzurollen, während die Graphiken darüber unverändert erhalten blieben.

03652 (0E44h) USR 3652
Eine Routine zum Löschen von Zeilen. Sie löscht die unteren 'x' Zeilen des Bildschirms, wobei der Wert von 'x' im B-Register enthalten sein muß. Der Bereich wird gelöscht und mit den momentanen Attributwerten gefüllt.

03756 (0EACh) USR 3756
Die COPY-Routine für ZX-Drucker. Gleich dem Befehl COPY. Ein Fehler von COPY ist, daß es nur die oberen 22 Zeilen des Bildschirms druckt und die beiden unteren Zeilen übergeht. Dennoch ist die Routine ansich sehr anpassungsfähig, falls sie richtig verwendet wird.
Wenn man USR 3756 direkt wählt, kommt es zu einem glatten Bildschirmkauderwelsch, aber bei Aufruf der Routine ab 3762 (0EB2h) wird es viel interessanter.
Diese Routine wird mit USR 3756 (0EACh) aufgerufen, was einige Variablen erstellt, um einen "normalen" Unrat darzustellen. Sie können die Variablen selbst erstellen, um einige nützliche Effekte zu erzeugen.

```
        DI           (Lebenswichtig!)
        LD B,192     Die Zahl der Bildschirmpixel,
                     die kopiert werden sollen)
```

Das ROM setzt dies auf 176, 22*8 Pixelzeilen. Wir können 192 Zeilen für eine komplette Bildschirmkopie setzen.

```
        LD L,16384   (Die Basisadresse der Bild-
                     schirmanzeige)
        CALL 0EB2    (Aufruf der Hauptroutine)
```

Es wäre möglich, die in HL geladene Adresse zu ändern, um auf einen Screen zu zeigen, der anderswo im Speicher abgelegt ist. Somit könnte man einen 'unsichtbaren' Screen COPYen!
Diese Routine endet mittels der ROM-Routine 0EDFh, die den Buffer des Druckers löscht. Dies ist unnötig, da die COPY-Routine diesen Bereich nicht verwendet. Wenn dieser Mißstand behoben worden wäre, hätte das ZX COPY auch auf 128K Maschinen funktioniert. Es ist möglich, einen ZX Printer oder Alphacom auf dem 128K oder dem grauen +2 zu betreiben, indem man die Routine übernimmt und den Aufruf zum Buffer löschen entfernt. Das Programm Fireview 2 tut dies.

Hier nun die Routinen, um über dem Bereich des Buffer zu drucken. Denken sie daran, daß diese Routinen nur in 48K anwendbar sind.

03789 (0ECDh) USR 3789

COPY mit Druckerbuffer. Diese Routine "kopiert" den Inhalt des Druckerbuffers (gespeichert bei 23296) logischerweise an den Drucker. Die Routine endet mit 0EDFh, was den Buffer, der von 23296 bis 23551 in Betrieb ist, löscht. Die Routine bei 0EDFh setzt auch die Systemvariable PR-CC (Drucker-Spalte) wieder auf Null und stellt Bit 1 von FLAGZ wieder her, um dem Computer mitzuteilen, daß der Buffer leer ist.

Vielleicht wollten sie schon immer mal wissen, was diesen "rumpelnden" Laut hervorruft?

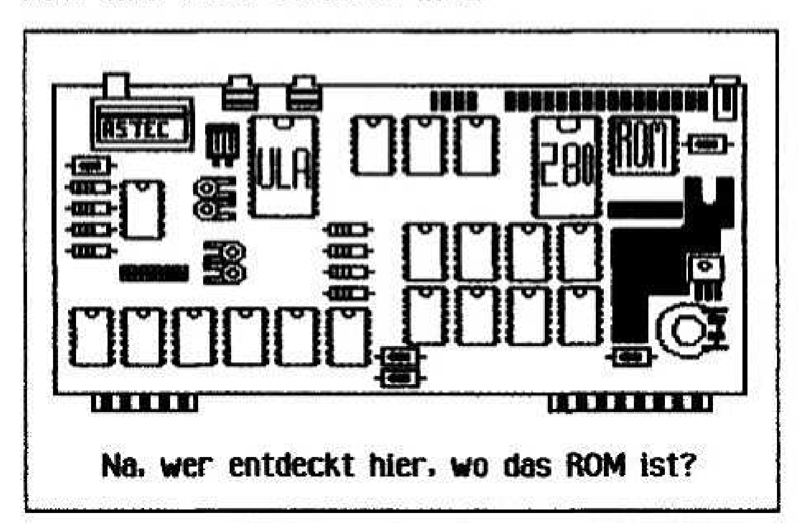
04223 (10F7h) USR 4223

ED Error. Dieses Stück Code ist wenig nützlich, aber sehr interessant. Es bewirkt das Summen, das Fehler begleitet, während Sie etwas eingeben. Wenn Sie den Bildschirm mit einer Zeile auffüllen oder Zeilen mit seltsamen Codes editieren wollen, können sie es hören.

```
10F7 BIT 4,(FLAGZ2)    (Falls nicht, dann kein
                       Summen)
     JR Z, 1026        (Kanal K benutzen)
     LD (ERR-NR),FF    (Fehler beheben)
     LD D,0            (hat die Länge des
     LD E,(RASP)       Summtons von den
                       Systemvars)
     LD HL,1A90        (Note gleichbleibend)
     CALL 0385         (BEEPer aufrufen)
```

04264 (10ABh) USR 4264

Eingabe über Tastatur. Eine ganz vernünftige Routine, die den Wert der momentan gedrückten Taste ins A Register zurückschickt. Komplexere Merkmale wie CAPSLOCK, PAPER und INK werden auch berücksichtigt. Das Carry-Flag wird gesetzt, wenn eine Taste gefunden wird.



Na, wer entdeckt hier, wo das ROM ist?

04535 (11B7h) USR 4535

Die Routine für 'NEW'. Sie kann von Basic oder MC aus verwendet werden. Trotzdem ist es leichter, NEW und ENTER einzugeben. Die Systemvariablem P-RAMT, RASP_RIP und UDG werden vernachläßigt, sowie auch der Wert im RAMTOP. Das gesamte RAM wird gelöscht, falls es unter dem RAMTOP liegt.
Diese Routine wird auch aufgerufen, wenn der Computer eingeschaltet oder mit USR 0 neu gestartet wird, was die NEW-Routine auf andere Weise bewirkt, indem sie den Computer glauben läßt, das RAMTOP bei 65535 liegt und das ganze RAM, die UDGs und Systemvariablen gelöscht werden müssen.

05808 (16B0h) USR 5808

Minimum setzen. Setzt den Editierbereich und alle Bereiche danach (Arbeitsspeicher und Rechnerstapel) auf deren kleinste Werte. Der bewirkte Effekt ist, das diese Bereiche 'gelöscht' werden.

05861 (16E5h) USR 5861

CLOSE#. Das A-Register enthält den Wert des zu schließenden Stromes. Dieses Codeteil läßt den Speccy abstürzen, falls Sie einen Kanal mit einem Wert von 4 oder darüber schließen, der nicht geöffnet war. Das liegt daran, das keine Markierung für das Ende in der Übersichtstabelle bei 1716h-171Ah ist. Das mußte per Hardwarezusatz im Interface 1 behoben werden. Es wäre leichter gewesen, dies bei neueren Spectrums einzubauen, wurde aber nicht gemacht. Deshalb wird beim Anwenden des Editors darauf hingewiesen, diesen CLOSE#-Befehl nicht zu benutzen, wenn kein Disklaufwerk verwendet wird.

05861 (1793h) USR 5861

Fehler im Interface 1. Der Gebrauch der Befehle CAT, ERASE, FORMAT und MOVE können den Speccy dazu bringen, die Fehlermeldung 'O Invalid

Stream' zu erzeugen. Dennoch wird dieser Fehler, falls ein Interface 1 angeschlossen ist, entdeckt und dazu verwendet, das IF 1 'aufzuwecken', sodaß es merkt, das es einen Befehl ausführen muß. Letzenendes sehen Sie den Fehler nie, da das IF 1 ihn beseitigt, sobald er passiert und dann den erforderlichen Befehl ausführt.

06229 (1855h) USR 6229
Eine Basiczeile listen. Diese Routine wird eingegeben, wenn HL die Adresse im Speicher hat, an der die Zeile eigentlich beginnt. Basic startet immer an der Adresse, die durch PEEK 23635 + 256*PEEK 23636 ermittelt wird. Dies ist die Systemvariable PROG. Es gibt eine ähnliche, die den Start der nächsten Zeile angibt.

06683 (1A1Bh) USR 6683
Anzeige der Zeilennummer. Dies zeigt eine Zeilennummer an, die in BC steht. Bedenken Sie, das Zeilennummern über 9999 nicht als richtige Dezimalzahl angezeigt werden.

06747 (1A5B) USR 6747
Zeigt eine Zahl über einen Strom an und läßt die vorgestellen Freiräume weg. Falls sie diese WIRKLICH wollen, sodaß alle Zahlen mit Freiräumen aufgefüllt werden, dann können Sie die Routine bei 1A28h aufrufen. HL muß auf die Adresse zeigen, wo die Zahl, die Sie anzeigen wollen, gespeichert ist. Dieses Beispiel schickt die Zahl 2500 an den momentanen Kanal:

```
LD BC,2500
CALL 1A5Bh
```

Die alternative Routine dazu sieht so aus:

```
LD HL,irgendeine Speicherstelle
LD BC,2500
LD (HL),BC
CALL 1A28h
```

Dies liefert die vorgestellten Freiräume, wie ich es beschrieben habe.

08855 (2297h) USR 8855
Ändern der Borderfarbe. Für diese Routine sollte eine Borderfarbe (0-7) ins A-Register übergeben werden. Sie ändert die Borderfarbe und setzt die Systemvariable vom Border auf den passenden Wert. Dies geschieht, damit sich die Borderfarbe nicht ändert, wenn der Lautsprecher im Einsatz ist. Eine kontrastierende Farbe für INK für die beiden INPUT-Zeilen wird ebenfalls eingerichtet.

08874 (22AAh) USR 8874
Routine für Pixeladresse. Wird sowohl von POINT als auch von PLOT verwendet und wird mit den Koordinaten eines Bildschirmpunktes in BC eingegeben. B enthält dabei den Wert für 'y' und C den Wert für 'x'.
Beim Verlassen enthält das HL-Registerpaar die Adresse des Bytes, das den Pixel enthält und das A-Register beinhaltet die Stelle des Pixels innerhalb des Bytes.

08911 (22CFh) USR 8911
POINT-Routine. Diese Routine muß mit der Pixelposition im BC Registerpaar eingegeben werden, genauso wie vorhin beschrieben. Der Wert, der normalerweise vom POINT-Befehl geliefert werden würde - in der Form: 0=Paper und 1=Ink - wird auf den Rechnerstapel gesetzt. Um ihn wieder zu erlangen, müßten Sie die Routine STK-TO-A bei Adresse 2314h aufrufen. Dann würde er im A-Register plaziert werden.

08927 (22DFh) USR 8927
Die PLOT-Routine. Einmal mehr muß die Position des Pixels, der geplottet werden soll, ins BC Registerpaar übergeben werden. Der Rest geschieht von allein. Der Status von INVERSE als auch von OVER wird berücksichtigt, so wie er in den Systemvariablen P-FLAG bei 23697 gespeichert ist.

08967 (2307h) USR 8967
Stapeln auf BC. Diese Routine nimmt ganz einfach einen 16-Bit Wert vom Floating-Point-Stapel und setzt ihn in das BC-Registerpaar.

08980 (2314h) USR 8980
Stapeln auf A. Wie zuvor erwähnt, kann dies jederzeit verwendet werden, um einen 8-Bit Wert vom Floating-Point-Stapel zu nehmen und im A-Register zu plazieren.

11400 (2C88h) USR 11400
ALPHANUM. Dies ist aber nun wirklich nützlich! Es wird mit einem Wert im A-Register eingegeben und falls es ein gültiger ASCII-Code für einen Buchstaben oder eine Zahl ist, wird das Carry-Flag gesetzt.
Falls es kein gültiger Code ist, wird das Carry-Flag zurückgesetzt. Es gibt zwei andere Routinen mit sehr ähnlicher Anwendung, und zwar:

11547 (2D1Bh),Zahl	(Spürt nur gültige Ziffern auf)
11405 (2C8Dh),Alpha	(Dies geht nur bei gültigen Buchstaben)

12457 (30A9h) USR 12457
Multiplizieren. Auch dies findet seine Anwendung. Es multipliziert HL und DE. Das Ergebnis erscheint im HL-Registerpaar. Dennoch wird im Falle eines Überlaufs (overflow) das Carry-Flag gesetzt.

Dies beschließt nun diese kurze Reise ins Spectrum ROM. In diesem Beitrag habe ich nur die Oberfläche der nützlichsten Routinen angekratzt. Es ließe sich schon alleine ein Buch über den Floating-Point-Kalkulator füllen!
Falls dies das Interesse an der Arbeit des Spectrum ROM geweckt hat, dann ist das einzige Buch, das ich empfehlen kann, das legendäre SPECTRUM ROM DISASSEMBLY von DR. IAN LOGAN und DR FRANK O'HARA.

Zur Betriebs-zuverlässigkeit unserer Computer

Eine Ergänzung

Zum behandelten Komplex ergaben sich noch einige Aspekte, die hier nachträglich zusammengefaßt werden sollen.
Manchmal findet man Hinweise, integrierte Schaltkreise elektronisch, d.h. mit Peltier-Elementen zu kühlen. Wenn der Gedanke auch nahe liegt, dem realen Einsatz stehen jedoch praktische Schwierigkeiten entgegen: Wie z.B. bei jedem Kühlschrank muß zum kühlen, so auch dem Peltier-Element, eine elektrische Leistung zugeführt werden, die dann zusätzlich abzuführen ist. In Groborientierung liegt diese zusätzlich benötigte Energie in Größenordnung der Energie, die man als Wärme abzuführen beabsichtigt. Also, ein auf einen Schaltkreis aufgebrachtes Kühlelement würde nur das Geräteinnere noch mehr aufheizen.
Selbst wenn ein fanatischer Hardware-Bastler ein Metallgebilde zur Wärmeabfuhr nach außen konstruieren würde, wäre das nächste noch schwierigere Problem nicht gelöst:
Wird nämlich das elektronische Bauelement nicht richtig gekühlt, etwa unter seine Umgebungstemperatur, dann entstehen in seiner Innenstruktur noch größere wechselnde Wärmegefälle als ohne Kühlung. Wie früher ausgeführt, ist das aber ein Hauptfaktor für die Alterung eines Bauelementes. Ein solcher einfacher Weg brächte so nur mehr Schaden als Nutzen. Eine elektronische Kühlung könnte nur dann vorteilhaft sein, wenn sie zeitlich geregelt, am Bauelement fühlergesteuert, erfolgte (Eine prima neue Einsatzmöglichkeit für einen ZX-Computer - oh je, in dem sollte man ja auch wieder kühlen!). Die Industrie jedenfalls läßt sich nur schwer vom simplen Lüfter abbringen.
Ja, und da gibt es auch für uns eine einfache hochwirksame Möglichkeit, in unseren ZX-Geräten für eine gute Innenkühlung zu sorgen, ohne irgendwelche Eingriffe vorzunehmen: Man kauft sich einen kleinen Computerstaubsauger, z.B. wie von (10) erhältlich. Statt der Saugdüse wird ein Plasteschlauch eingesteckt, sein anderes Ende wird durch den Bus-Schacht in den Computer eingeführt. Geht es dort durch Peripherie zu eng zu, müßte man ggf. doch noch ein Loch für den Schlauch bohren. Der Luftdurchsatz und so die Abkühlung im Gerät werden so um ein Vielfaches verbessert, als durch jede andere Maßnahme, die als Möglichkeit früher aufgeführt wurde. Den Motor betreibt man zweckmäßigerweise mit einem (oft schon vorhandenen) Miniatur-Netzteil.
In (11) wurden noch ein paar Tips gegeben: Um die Luftzufuhr zur Computer-Unterseite - hauptsächlich betreffend ZX81 und Gummispectrum - zu verbessern kann man auf die vier Gummifüße nochmals solche von 2-3 mm Dicke aufkleben (Auf keinen Fall sollte man die vorhandenen, etwa nach einer Gehäuseöffnung oder weil einer verlustig gegangen ist, weglassen oder entfernen).
Die dort für den ZX81 angeführte Vergrößerung des Kühlbleches des Spannungsreglers ist für letzteren auf jeden Fall von Vorteil. Die Wärmemenge im Gerät wird aber dadurch nicht reduziert.
Wie schon früher angeführt, ist hier die Temperaturdifferenz zwischen dem heißen Objekt - dem Kühlblech - und dem empfangenden Medium - dem Gehäuse - noch zu klein als daß der durch Strahlung abgegebene Anteil der Wärmemenge von Bedeutung ist. Das heißt auch, eine Oberflächenschwärzung bringt kaum merkliche Vorteile (Deshalb brauchen wir auch unsere Heizkörper im Wohnraum nicht schwarz anzumalen. Das wäre sonst vielleicht auch ein neuer Wohnkultur-Modetrend? Ihre Konstruktion ist vielmehr auf gute Konvektionsleistung ausgerichtet).
Dazu sei noch bemerkt, daß eine Eloxalschicht zwar hilft, den Wärmestrahlungsanpassungswiderstand etwas zu verbessern, wesentlich optimaler sind aber Beschichtungen, die amorphen Kohlenstoff (Ruß) oder Schwarzchromate enthalten, wie sie z.B. bei Solarenergiewärmeaustauschern Einsatz finden.
Übrigens, eine schwarze Oberfläche ist schwarz, weil sie die für uns im elektromagnetischen Strahlungsbereich sichtbaren Lichtstrahlen absorbiert. Das heißt aber durchaus noch nicht, daß sie auch die infraroten Wärmestrahlen gut aufnimmt. So gibt es z.B. Schwarzglas ultraviolettdurchlässig oder auch infrarotdurchlässig (Verwendet bei Strahlungsschranken oder in der technischen Photographie). Also schwarze Materie muß nicht immer ein besserer Wärmeabsorber sein als andersfarbige.
Von einem Aufrauhen der Spannungsreglerkühlblechoberfläche möchte ich aber abraten. Das erhöht nur ihren Oberflächenströmungs-Widerstand, der für gute Konvektion ja gering sein muß.
Die früher als Grobschätzung angenommenen Werte für die Ausfallrate der Sinclair Computer lassen ersehen, daß ihre Zuverlässigkeit nicht besonders hoch ist. Im Laufe der Geräteentwicklungen bis zum Spectrum +3 und dem ZX88

(12), (13) zeichnet sich zwar eine Verbesserung ab, die aber noch nicht für Einsatzzwecke ausreicht, die hohe Zuverlässigkeit und Sicherheit erfordern, zumal dafür noch in die Geräte Eingriffe gemacht werden müssen. So erfordern z.B. Objekte wie die oft zitierte Heizungssteuerung oder Objektüberwachung den Einbau eines Autostarts, der das System nach Netzausfall wieder einrichtet; der Wackel-Bussteckverbinder ist den Andorderungen nicht gewachsen und und -.
Wir wollen also unsere Computer in den Einsatzbereichen belassen, wofür sie gedacht sind bzw. wofür sie sich in sonstigen Anwendungen auch eignen oder auch mit ihnen ein paar uns interessierende Experimente machen. Die jeweils für einen speziellen technischen Einsatzzweck entwickelten und produzierten Geräte und Systeme erfüllen ihre Aufgabe weitaus unkomplizierter und sicherer.
Allgemein zur Perspektive der Zuverlässigkeit von Computern und Computersystemen zeichnet sich eine erstaunliche Steigerung ab. Gegenüber den für die bisherigen Sinclair-Geräte angenommenen Werte hat man so z.B. in Hochsicherheits-Computersystemen im Bereich Raumfahrt und Militärtechnik teilweise mit Einsatz redundanter Schaltungstechnologie heute schon Verringerungen der Ausfallraten bis um 4 Größenordnungen erreicht. Das heißt, Ausfälle sind schon sehr seltene Ausnahmen.

Quellennachweis

(10) Computerstaubsauger im Conrad Katalog '96, S. 1042
(11) Tips aus Dresden; ZX-Team-Magazin 3/95, S. 21
(12) Neues in Kürze; SUC Infoheft 12/94, S. 3
(13) Der ZX88, Sir Clives Wunderding, SPC-Infoheft 7/95, S. 9

Zuletzt noch eine kleine 'Dreckfühler' Korrektur:
- Heft 8/95, S. 11, linke Spalte, 2. Absatz, 2. Zeile von unten: Edustreuungen -> Einstreuungen.
- Heft 9/95, S. 7, rechte Spalte, 17. Zeile und Heft 10/95, S. 15, linke Spalte, letzter Absatz: Elektrizitätsbereich -> Elastizitätsbereich.
- Heft 11/95, S. 7, rechte Spalte, vorletzter Absatz am Ende: ... in dieser Höhe **nicht** zur Verfügung steht.

An dieser Stelle möchte ich mich noch einmal bei allen bedanken, die mir Zeilen zur Fortsetzungsserie zukommen ließen. Ich habe versucht, einige Dinge anschaulich, manchmal an Beispielen aus anderen technisch-physikalischen Bereichen, zu erläutern und Hinweise zu geben.
Jedenfalls habe ich mich gefreut, daß solche Beiträge, wie auch die Meinungen in den Info-Heften zeigen, nicht nur gelesen werden, sondern auch interessieren und Resonanz gefunden haben.
Hier auch meine herzlichen Dank dem WoMo-Team für seine unermüdliche redaktionelle, schriftsetzerische, Graphik-erfindende und organisatorische Arbeit. U.a. mußte doch auch so manche Seite handschriftlicher Text ins Druckbild übersetzt werden und wurde noch mit einem kleinen Scherzbild versehen.
Gratulation zur 75sten Ausgabe des Informationsheftes! Eine tolle Leistung!

Heinz Schober, Taubenheimer Straße 18
01324 Dresden



Wenn man einmal über 300 bespielte Disketten hat, die meisten davon 720 KB, wird es schwierig, ein länger nicht benutztes Programm oder gar eine kleine Routine zu finden und noch zu wissen, wie man sie benutzt. Katalogprogramme, auch ausführliche, können da wenig aussagen und bevor man gedruckte und "gutverwahrte" Anleitungen wiederfindet, vergeht allerhand Zeit. So kommt es auch vor, daß man die Kataloge sämtlicher Disketten 2-3mal anschauen muß, bis man etwas wiederaufindet, auch wenn es doppelt gespeichert war.
Auch sogenannte Directories, die es ja auch gibt, mit denen man auf einer Diskette z.B. 4 Kataloge, auch mit gleichem Namen einrichten kann, zeigen ja nur Namen, sagen aber nichts über die einzelnen Programme aus.
So kam mir er Gedanke, jede Diskette mit einer eigenen, internen Datei zu versehen, in der man das Gewünschte suchen kann und gleich das Wichtigste darüber erfährt, es auch gleich von da laden kann.
Das Programm habe ich vom englischen SDC Club, der leider seinen Geist aufgegeben zu haben scheint, geschrieben von C. Robotham, übernommen, es heißt dort "composer", und textlich leicht verändert um es diesem Zweck anzupassen.
Es ist eine allgemein verwendbare, sehr einfache und angenehme Datei, z.B. auch für CD's, MC's, LP's oder für Bücher usw.
Es passen etwa 400 Datensätze zu 82 Buchstaben hinein (32.8 KB), die in einer zweidimensionalen Stringvariablen gespeichert sind. Man kann aber

auch mehr Buchstaben pro Datensatz dimensionieren, dann gibt es entsprechend weniger Datensätze.
Hier habe ich sie als interne Diskettendatei, die zusätzliche Informationen über die auf der Diskette gespeicherten Programme, Routinen und Files gibt, eingerichtet.
Die schnelle Suchfunktion findet in kürzester Zeit jeden Begriff, jedes angefangene oder bruchstückhafte Wort, gibt Auskunft über die Möglichkeiten jedes Programms und erlaubt dieses, soweit es sich um ein Basic- oder Basicladeprogramm handelt, zu laden. MC-Routinen können erklärt werden, sodaß ihre Funktion und Einsatzmöglichkeit klar wird. Wenn man sehr viele Programme hat, ist es wohl die einzige Möglichkeit, alle im Auge zu behalten und keins, sowie deren Einsatzmöglichkeit zu vergessen.
Die Handhabung ist so einfach, das Programm selbsterklärend, das es wirklich von allen mir bekannten Dateien die am wenigsten anstrengende ist. Eingeben muß man die Dateien allerdings auch hier, das bleibt einem bei keinem Programm und keinem Computer erspart, so groß und modern er auch immer sein mag.
Man läßt sich zweckmäßigerweise zuerst einen Catalog der Diskette ausdrucken, möglichst einen mit Angaben ob Basic, MC oder DAT File und beschreibt dann der Reihe nach die einzelnen Programme in der Datei. Was man nicht gleich weiß, kann man ja später nachtragen und ergänzen. Wieviele Einträge man will, hängt ganz davon ab, wieviele Files auf der Diskette sind oder noch gespeichert werden sollen. Für alle Nebnroutinen braucht man aber nicht je ein File, wenn sie nicht extra erklärt werden sollen, denn sie werden ja meist automatisch mitgeladen.
VORSICHT beim Absaven! Wenn eine falsche Diskette im Laufwerk ist, werden deren Diskettendaten überspielt und stimmen dann nicht mehr. Abhilfe: Datenfile nicht bei jeder Diskette nur "D" nennen, sondern nummerieren oder abgekürzten Diskettennamen verwenden.

Herbert Hartig, Buchloe

Interview





WoMo: Hallo LCD. Viele kennen Dich ja eigentlich nur unter Deinem Pseudo "Tigers Claw". Wie bist Du eigentlich auf diesen Namen gekommen?

LCD: Ich stand vor der Qual der Wahl. Es gab mehrere Pseudos zur Auswahl, Tiger's Claw stammte ursprünglich aus dem PC Spiel Wing Commander, das mir ziemlich gefiel. Ich wollte etwas das mich gut beschreibt und zu mir paßt. Ich will damit nicht sagen, das es keine passenderen Pseudos gab. Damals fand ich, es sei eine gute Idee mich so zu nennen, inzwischen sind ja mehrere Jahre vergangen und mein Charakter hat sich auch geändert, heute würde ein anderes Pseudo passender sein. Damals war ich eben etwas "schärfer" als jetzt.

WoMo: Seit wann hattest Du konkret den Gedanken, einen Club in Österreich zu gründen? Wo es in ganz Österreich eigentlich garnicht so viele Spectrum-User mehr gibt, ganz zu schweigen von SAM-Usern? Wie kommt man generell auf so eine Idee?

LCD: ASC war schon vor acht Jahren als Idee geboren worden, damals hatte ich aber weder die Erfahrung, noch die Connections, die ich heute habe, um diese Idee zu verwirklichen. Diese Idee wurde dann vorläufig fallengelassen, um jetzt endlich umgesetzt zu werden.
Ich war vor einiger Zeit Mitglied im Chipsi Computer Club, hier in Wien, als es dort noch einige Spectrum User gab, nur jetzt sind nur noch Amiga und PC User da. Soviel ich weiß, müßte es auch in Wien noch viele Spectrum User geben, denn auch wenn die User ihre Geräte verkaufen, muß es ja doch Leute geben, die sie wiederum kaufen. Man kommt auf noch verrücktere Ideen als meine, wenn man um die Szene fürchtet. Der Szene kann man wohl am besten helfen, wenn man einen Sammelpunkt, einen Club gründet, in dem der User merkt, das er nicht alleine ist, das es noch viele Gleichgesinnte gibt, mit denen er Erfahrungen und Ideen austauschen kann.

WoMo: Was soll bei Dir überwiegen: PD Software für den Spectrum oder eher für den SAM?

LCD: Diese Frage ist im Augenblick noch schwer zu beantworten. Noch überwiegt eindeutig die ZX Spectrum Software, aber wie es in Zukunft aussehen wird, kann ich nicht beantworten. Alles hängt davon ab, was ich so hereinbekomme. Mit Sicherheit kann ich nur folgendes sagen: In Wien gibt es außer mir noch einen SAM Coder, ich muß nur herausfinden, ob sein SAM wieder in Ordnung ist. Bisher habe ich nur ein einziges Programm von ihm erhalten, das kann doch wohl nicht alles gewesen sein, oder? Ich selbst habe auch einiges für den SAM geschrieben und habe auch vor, dies mal öfters zu wiederholen.

WoMo: Wie siehst Du die Zukunftschancen für die beiden Rechner?

LCD: Also meiner Meinung nach, wird sich der SAM auf lange Sicht klar durchsetzen, vor allem weil in England so viel für ihn getan wird. All die Shows, Hardware und Softwareneuentwicklungen hätten wohl nicht stattgefunden, wenn der SAM nicht immer populärer geworden wäre. Der Spectrum hat sehr wohl auch Chancen, das Jahr 2050 zu überleben, und nicht nur als extrem billiger Steuercomputer, sondern auch als die wohl beste Maschine außer SAM, um Basic zu lernen, und der wohl einzige virenfreie Computer. Auch als billiger Babysitter (mit Jet Pac) wird der Spectrum eine harte Fangemeinde haben, die immer neue Software programmiert. Um ehrlich zu sein, verstehe ich einige Leute nicht, die von Spectrum auf den PC umgestiegen sind, denn hier sind sie immerhin etwas Besonderes, während sie auf dem PC nur einige unter hunderttausenden sind. Mit der Unterstützung von Autoren, die ihre Programme als PD kostenlos weitergeben, oder sie auch verkaufen, kann der Spectrum wieder aufblühen, jedoch muß da jeder mithelfen, der es kann.

WoMo: Welche Kooperationen hat der ASC zu anderen Clubs und zu welchen? Wie beurteilst Du die Zusammenarbeit mit dem SPC?

LCD: Klarer Fall, mit dem SPC stehe ich in allerbester Beziehung. Ich meine, der Programm-austausch geht gut voran, ich schreibe manchmal Artikel für Euch, schicke selbstgeschriebene Programme als erstes an SPC u.s.w.
Mit anderen Clubs ist es schon anders, ich meine nicht, daß wir uns an die Kehle springen, sondern das ich meistens in Briefkontakt mit denen stehe. Spectrum Club Ungarn hat für mich sogar eine kurze Artikelserie über den Z87 Coprozessor für Spectrum geschickt, den SUC auch abdruckt (keine Ahnung, ob SPC es nicht auch tut). An SUC schicke ich meine selbstprogrammierten Spiele, um zu sehen, was Thomas darüber schreibt, und nicht selten bekomme ich dadurch noch mehr Kontakte, als wenn ich mich wieder an Demos versuchen würde. Mit SGG stehe ich im Magazinaustausch, obwohl ich gewisse Probleme mit der Sprache habe. Dann gibt es noch seit nicht allzulanger Zeit Kontakte zu FPD, das zwar kein Club ist, aber dennoch ziemlich interessant. Ich habe auch vor, weitere Kontakte nach England, Tschechei und Polen zu knüpfen. Mit anderen Worten: Ich bin voll dabei, den ganzen Kontinent nach Clubs abzuklappern, die etwas mit Speccy oder SAM zu tun haben.

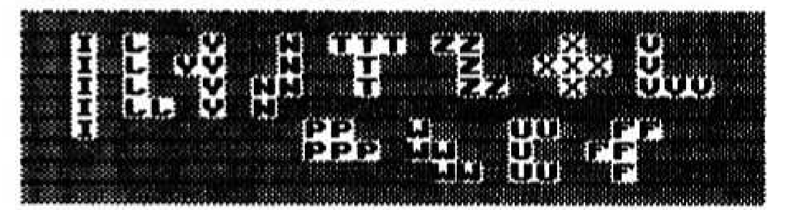
WoMo: Welche Programme können wir in Zukunft von Dir erwarten? Wirst Du auch weiterhin auf PD setzen, oder planst Du langfristig eine Vermarktung Deiner Produkte?

LCD: Also, in Planung ist ein neues Demo, natürlich mit ASC Werbung. Dieses wird eher etwas für Erwachsene sein. In der Mache ist bereits "Legacy of the Zer Empire", das eindeutig kommerziel sein und z.Z. nur auf +D erscheinen wird. Premiere dabei ist, das es wahrscheinlich über drei Disks einnehmen wird und in Deutsch erscheint, die englische Version folgt dann später. Mit demselben System wie "Legacy", will ich auch noch weitere, weniger komplexe Megaadventures erschaffen, die dann in meinem Diskmag erscheinen werden. An konkreten Titeln kann ich folgende nennen: X-Files (Akte X), Married with Children (Eine schrecklich nette Familie) und Frankenstein Unbound. Dann habeich noch Pläne für ein neues Patience Spiel (Pyramid Patience, PD), Strategie Kriegsspiele: Unknown Danger und Starfleet Battles. Mein altes Projekt Star Trek Classic wird wieder hervorgeholt, aber davon wird es nur eine 128K Version geben, genauso wie die Wirtschaftssimulation World Shiping. Dann muß noch mein DTP Programm verbessert werden, so gesehen, bin ich bis zu meinem natürlichen Lebensende ausgebucht.
Mit meinem neuen Amiga zu Spectrum Picture Converter habe ich es jetzt viel einfacher, Graphiken für Programme herzustellen, das beschleunigt den Programmiervorgang um einige Prozentteile und sieht auch viel besser aus als wenn ich es gezeichnet hätte. Der Converter ist auch PD, aber da AMOS Programme notwendig sind um die Bilder umzurechnen, sind die beiden Disks z.Z. nur über micherhältlich.
Längerfristig habe ich vor, ähnlich wie FRED auch Spectrum und SAM Produkte anderer Coder zu vermarkten, aber das soll nicht bedeuten, daß ich dann keine PD mehr erzeuge. Wenn Programme meine Erwartungen nicht erfüllen, gehen sie als Freeware auf das Diskmagazin, das wird sowohl der Freeware wie auch den kommerzielen Projekten zugute kommen, denn ich kann mir kaum vorstellen, daß jemand zwanzig Mark für ein Programm bezahlt, wenn es ein ähnliches im PD Bereich gibt. Das erhöht den Qualitätsdruck ganz erheblich. Sicher ist, daß die Programme, die ich verkaufen werde, weder Pac Man noch Space Invader Clones sein werden.
Es wird auch Exclusivprogramme geben, die nur für SCENE gemacht wurden und trotzdem keine PD sein werden, die Autoren werden dafür eine kleine Belohnung kassieren, ähnlich wie bei FRED. Was den SAM angeht, so werden alle meine Programme PD sein, sofern nicht anders vermerkt.
Das war jetzt eine zu ausführliche Antwort, sorry deswegen, ich hoffe daß ich den anderen Artikelschreibern nicht allzu viel Platz weggenommen habe.

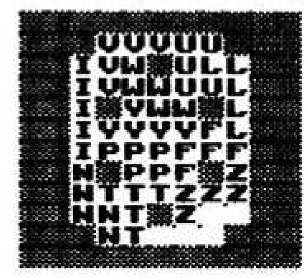
WoMo: Wir bedanken uns bei Dir für dieses wirklich interessante Interview! Viel Erfolg für die Zukunft.

PUBLIC DOMAIN made in Germany

Pentamino (Angrosoft)
Manchmal erreichen uns ganz überraschende Programme von Clubmitgliedern, wie z.B. Pentamino von Eduard Bröse. Pentamino? Ja, was ist das denn? Die Erklärung lieferte uns der Autor gleich mit:
Mit diesem Programm kann man "Pentamino" Rätsel lösen. Es findet heraus, wie man zwölf verschiedene Teile (aus je 5 Zellen)

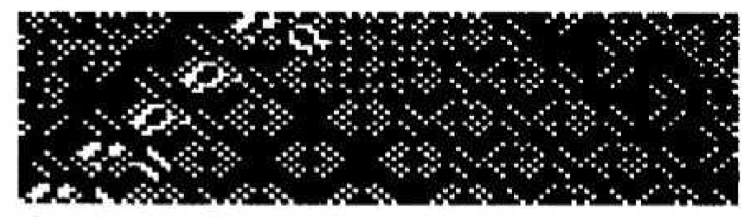


zusammenlegen kann, um eine eingegebene Figur wie im abgebildeten Beispiel zu bauen.

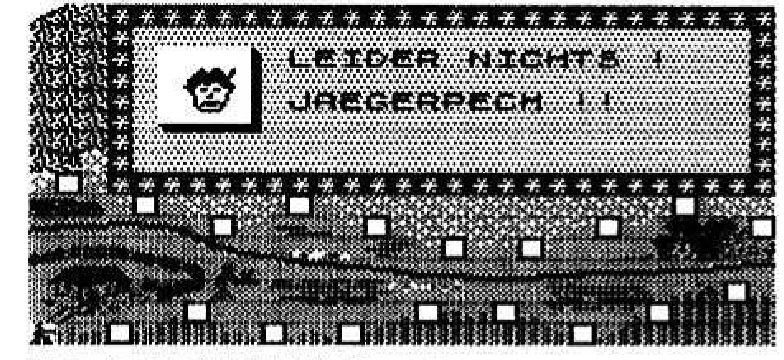


Eine Anmerkung von uns: So ganz ist uns das nicht gelungen, wie man sieht. Aber die Lösung ist dennoch klar ersichtlich. Jetzt suchen wir natürlich nach weiteren "Pentamino"-Figuren.

Life (Angrosoft)
ist die berühmte Imitation des "Lebens" und der "Genesis". Die Imitation läuft auf einem Feld in der Größe von 112 x 88 Zellen (je von 2 x 2 Pixeln).
Nun - wir denken, das die Regeln des "Life" eigentlich bekannt sind, es gibt auch schon etliche Versionen davon. Was dieses zweite Programm von Eduard Bröse so interessant macht, ist die "Genesis" Version, sowie einige Möglichkeiten, das Programm zu beeinflussen, z.B. durch Einsatz eines Virus.
Zu jedem der beiden Programme ist auch ein Assembler-Quelltext für ZEUS beigefügt.

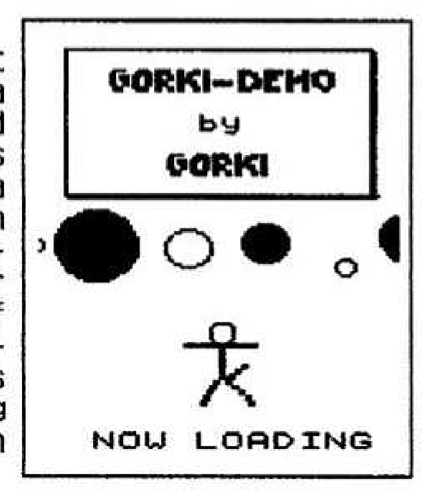


Geordnetes Chaos bei der "Schöpfung" (Genesis)

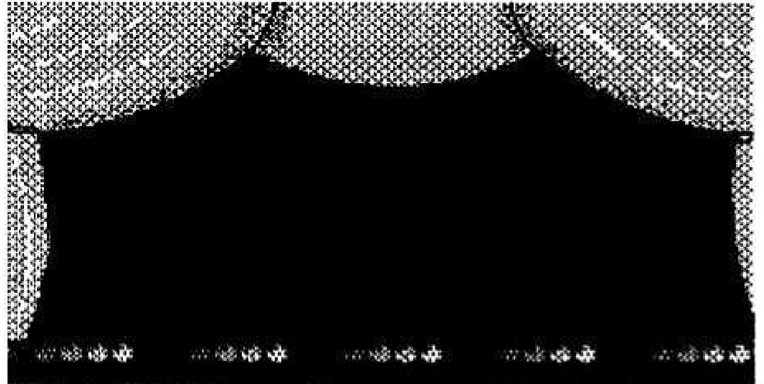


Das Jagdspiel (WoMo-Team)
Eigentlich wollten wir über dieses Spiel den Mantel des Schweigens hüllen. Andererseits mag die Welt schon schlimmeres gesehen haben. Es handelt sich um ein "friedliches" Jagdspiel, bei dem nur mit Würfeln geschossen wird. Jedes Tier hat eine andere Wertigkeit, was sich auf die Punktzahl auswirkt. Es gibt auch Tiere, auf die man nicht schiessen (würfeln) darf, z.B. Eichhörnchen. Auf jeden Fall geht es hier friedlicher zu, als auf einer richtigen Jagd und fordert (hoffentlich) nicht die Tierschützer auf den Plan. Ach ja, wie bei fast allen Spielen von uns kann der Spectrum hier bis zu 6 Spieler ersetzen, also komplett gegen sich selber spielen.

Gorki-Demo (Gorki)
Dieses Demo dreht sich um das Thema "Sex in der Kunst", und ist ein Debut des Programmierers. Da dieses Demo auf einem reinen 48er geschrieben wurde, hat der Programmierer auf musikalische Untermalung verzichtet. Es hätten aber ruhig einige Bilder mehr sein können.



Skeleton Dance (Rüdiger Döring)
Dieses Demo war jahrelang als "unvollendetes secret" auf einer unserer Disketten. Rüdiger wollte es so nicht freigeben. Jetzt hat er die Spectrum-Szene verlassen, und wir haben uns seine Genehmigung eingeholt. Es wäre auch zu schade gewesen. Einfach sehens- und hörenswert! Noch ist die Bühne leer...



PUBLIC DOMAIN

by Miles Kinloch

Heute mal einige Plus D Utilities von Miles Kinloch zum Eintippen. Es lohnt sich! Die Programme erklären sich von selbst.

```
  10 REM FAST FILE CONVERTER FOR
  20 REM TASWORD 128 > TASWORD 2
  30 REM (PD) Miles Kinloch 1995
  40 REM
  50 CLEAR 31899: GO SUB 1000
  60 INPUT "Load from which drive? ";s:
IF s<>1 AND s<>2 THEN  GO TO 60
  70 INPUT "Save to which drive? ";d:
IF d<>1 AND d<>2 THEN  GO TO 70
  80 POKE 23728,1: CAT s;"????????_T"!:
RANDOMIZE USR 8: POKE 23728,0: LET
n$="(8xSpace)_T": INPUT "Filename? ";
LINE n$( TO 8)
  90 CLS : CLEAR #: OPEN #5;ds;n$IN
 100 RANDOMIZE PEEK 23631+256*PEEK
23632+PEEK 23584+256*PEEK 23585+30:
POKE 31915,PEEK 23670: POKE 31916,PEEK
23671: POKE 23729,0
 110 LET end=USR 31900: CLOSE #*
 120 SAVE dd;n$( TO 8)CODE 32000,end-
32000: STOP
1000 LET c=0: FOR n=31900 TO 31983:
READ d: LET c=c+d: POKE n,d: NEXT n
1020 IF c<>8178 THEN  PRINT "ERROR
 IN DATA!": STOP
1030 RETURN
2000 DATA 62,5,205,1,22,30,0,33,0,
2010 DATA 125,34,254,124,237,75,0,
2020 DATA 0,120,177,68,77,200,213,
2030 DATA 229,205,230,21,225,209,
2040 DATA 254,13,32,10,87,58,177,
2050 DATA 92,183,122,40,15,24,32,
2060 DATA 254,10,40,219,119,35,28,
2070 DATA 175,50,177,92,24,210,123,
2080 DATA 254,64,32,8,175,95,60,50,
2090 DATA 177,92,24,197,54,32,35,28,
2100 DATA 24,237,6,64,54,32,35,16,
2110 DATA 251,24,182
9999 SAVE d1"TAS128TO2" LINE 10
```

```
  10 REM +D G+DOS/BETADOS TEST
  20 REM (PD) By Miles Kinloch
  30 REM
  40 REM 20 bytes relocatable code.
Call with LET x=USR <start>. x will be
0 for G+DOS and 1 for Betados. (RUN
1000 to test.)
  50 CLEAR 39999
  60 FOR a=40000 TO 40019
  70 READ d: POKE a,d: NEXT a
  80 DATA 219,231,58,172,48,1,0,0,254,
68,40,4,254,205,32,1,12,211,231,201
  90 PRINT "PRESS ANY KEY TO SAVE
CODE.": PAUSE 0: SAVE d1"WHICHDOS"CODE
40000,20
 100 CLS : STOP
1000 REM Test
1010 CLS : PRINT "DOS INSTALLED:
";"BETADOS" AND USR 40000;"G+DOS" AND
NOT USR 40000
```

```
  10 REM +D LAST DRIVE NO. TEST
  20 REM (PD) By Miles Kinloch
  30 REM
  40 REM For +D/Betados/G+DOS only.
(Not for Unidos or Disciple.) Call
with LET x=USR <start>. x will be the
drive last used. (RUN 1000 to test.)
  50 CLEAR 39999
  60 FOR a=40000 TO 40014
  70 READ d: POKE a,d: NEXT a
  80 DATA 219,231,58,218,61,1,1,0,31,
56,1,12,211,231,201
  90 PRINT "PRESS ANY KEY TO SAVE
CODE.": PAUSE 0: SAVE d1"WHICHDRIVE"
CODE 40000,15
 100 CLS : STOP
1000 REM Test
1010 CLS : PRINT "CURRENT DRIVE: ";USR
40000
```

Alle genannten Programme können auch über unsere PD bezogen werden.

```
+D  SNA FILE LOADER by R. SWANN
(I think), modified by EWGF.

 S N A P   R E A D E R   V . 4

    This program is FREEWARE

IT CAN BE COPIED AND DISTRIBUTED
BY ANYONE OTHER THAN PRISM PD

MARTYN SHERWOOD, THIS MEANS YOU.

          Press any key
(unless you're martyn sherwood
     in which case press RESET)
```

Last but not least noch ein Programm, an dem Miles Kinloch maßgeblich mit dran gearbeitet hat. Es heißt "SNA READ 4a" und ist in der Lage, PC Z80 Emulator Snapshots per Plus D einzulesen. Das Programm ist sehr gut dokumentiert. Ob der "nette Gruß" im Ladescreen nun wirklich nötig war, bleibt denen überlassen, die das Programm gemacht haben.

The *Spectrum & SAM* Bulletin Board

published by:
Harald R. Lack, Heidenauerstr. 5, 83064 Raubling / Hubert Kracher, Starenweg 14, 83064 Raubling

Red Moon

Liebe Mituser!!

Die Vorarbeiten haben wir also jetzt hinter uns und wir hoffen, daß Ihr Euch ein wenig mit dem Plan vertraut gemacht habt. Dann kann es jetzt also losgehen. Wir stehen an unserem Ausgangspunkt - on a wide, flat grassy plain - und machen wie folgt weiter:
Dig (wir finden eine Krone), N, NW, NW, NW, N, take gloves, S, SE, W, take dagger, E, SE, W, W, S, take horseshoe, examine horseshoe (fühlt sich irgendwie magisch an, ist aus Eisen und magnetisch), SE, E, S (wir sind jetzt an der Ruine eines Steinhauses, einem idealen Versteck für unsere Gegenstände und dem Platz, an den am Ende der Mondkristall gebracht werden muß), take lamp, light lamp, drop crown, drop dagger, drop gloves.
Wenn wir jetzt Score eingeben sollte etwa folgender Text erscheinen: Score 100/1000 and you are a winging macod. Der Score ist natürlich davon abhängig, ob wir noch einige andere Aktionen geliefert oder ob wir den Stand zwischenzeitlich abgespeichert haben, da dies alles einen Einfluß auf unsere Hitpoints und damit auf den Score hat. Solange jedoch keine gigantischen Abweichungen auftreten, hat dies keinen Einfluß auf die erfolgreiche Adventurelösung. Doch kommen wir nun zu unserem weiteren Vorgehen. Wir sind ja noch in dem ruined stone house und machen jetzt folgende Schritte:
N, E, E, S, E, N, N, examine bushes (wir finden eine Kurbel), insert handle in square hole (paßt zufälligerweise ganz genau), turn handle (die Schleusen öffnen sich und der See wird trocken gelegt), N, examine fungus (wir finden ein Perle), N, E, N, bury rat, N, take dulcimer, E, E, E, open door, D, E, S, S, bury sog, E, S, S, open door, S, open door, W, S, W, SW, NW, take bracers, wear bracers (dadurch können wir jetzt bis zu 12 Gegenstände tragen), take shirt, SE, SW, SW, W, wave horseshoe (der Magnet zieht die Münzen an), E, NE, NE, N, N, E, NW, bury bletch, N, open door, N, W, S, NW, W, W, SW, unlock gate, with key (das Tor öffnet sich und der Schlüssel zerfällt zu Staub), SW, S, W, W, S, drop horseshoe, drop coins, drop pearl, drop shirt.
Wir sollten jetzt ca 250/1000 haben und ein Real Adventurer sein.
Gehen wir jetzt auf die dritte Entdeckungsreise.....
N, W, S, say humak (die Türbolzen klappen zurück), open door, in, take book, D, D (Xiiz will uns ohne eine Gabe nicht vorbeilassen), give book (wir können uns das Buch später mittels eines Zaubers wieder zurückholen), D, S, SE, SW, take medallion, say satarh (dadurch gelangt das Buch, das wir Xiiz gegeben haben, in das alte Steinhaus), NE, NW, N, U, U, U, say humak, open door, out, N, E, S.
Zurück im alten Steinhaus: take book, read book (darin steht OBIS to open and Ollabin to dust; wir brauchen das viel später, um mit der Mumie einen Handel zu machen), drop book. Bei Score erhalten wir jetzt ca. 350/1000 und wir sind ein Real Adventurer.
Soviel für heute. Nächstesmal schliessen wir dann hier an. Inzwischen viel Spaß beim Ausprobieren und Nachspielen.....

ANZEIGEN